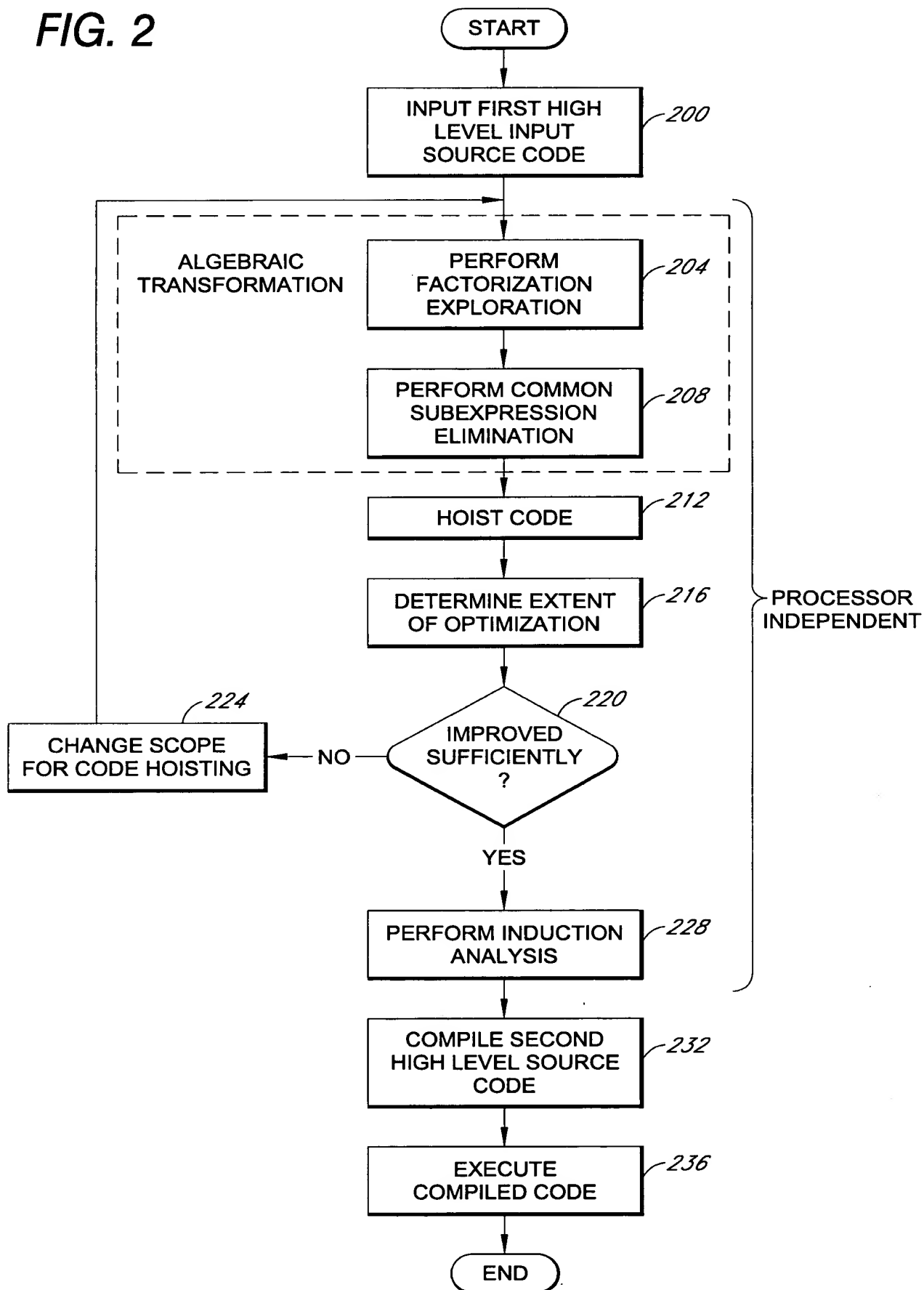


FIG. 1

FIG. 2



09760129-0514-01

```

for (y=0; y<M+3; ++y) {
  for (x=0; x<N+5; ++x) {
    ...
    if ((x-3)>=1 && (x-5)<=N-2 && (y-2)>=1 && (y-3)<=M-2) {
      if ((x-5)>=1 && (y-3)>=1 {
        if (out_compute == 255) {
          if (comp_edge_pixels [ ((x-4) %3) *3+ (y-2) %3] <comp_edge_middle) out_compute=0;
          ...
          if (comp_edge_pixels [ ((x-4) %3) *3+ (y-4) %3] <comp_edge_middle) out_compute=0;
          ...
          if (comp_edge_pixels [ ((x-5) %3) *3+ (y-4) %3] <comp_edge_middle) out_compute=0;
          ...
        }
      }
      if ((x-3)<=N-2 && (y-2)<= (M-2)) {
        maxdiff_compute =
          max13 (abs (gauss_xy_pixels [ ((x-2) %3) *3+ (y-1) %3]
                    - gauss_xy_middle), maxdiff_compute);
          ...
        maxdiff_compute =
          max13 (abs (gauss_xy_pixels [ ((x-2) %3) *3+ (y-3) %3]
                    - gauss_xy_middle), maxdiff_compute);
          ...
        maxdiff_compute =
          max13 (abs (gauss_xy_pixels [ ((x-3) %3) *3+ (y-3) %3]
                    - gauss_xy_middle), maxdiff_compute);
          ...
      }
    }
    ...
  }
}

```

FIG. 3

FIG. 4

```

for (y=0; y<M+3; ++y) {
  for (x=0; x<N+5; ++x) {
    ...
    if (x>=4 && x<=N+3 && y>=3 && y<=M+1) {
      if ((x-5)>=1 && (y-3)>=1) {
        if (out_compute == 255) {
          csexmin4mod3x3 = ((x-4)%3)*3;
          cseym4mod3 = (y-4)%3;
          if (comp_edge_pixels[ csexmin4mod3x3 + (y-2)%3 ] < comp_edge_middle) out_compute=0;
          if (comp_edge_pixels[ csexmin4mod3x3 + cseym4mod3 ] < comp_edge_middle) out_compute=0;
          if (comp_edge_pixels[ ((x-5)%3)*3 + (cseym4mod3) < comp_edge_middle) out_compute=0;
          ...
        }
      }
      if ((x-3)<=N-2 && (y-2)<=M-2) {
        csexmin2mod3x3 = ((x-2)%3)*3;
        cseymod3 = y%3; /* = (y-3)%3 */
        maxdiff_compute =
          max13(abs(gauss_xy_pixels[ csexmin2mod3x3 + (y-1)%3 ]
            - gauss_xy_middle), maxdiff_compute);
        ...
        maxdiff_compute =
          max13(abs(gauss_xy_pixels[ csexmin2mod3x3 + cseymod3 ]
            - gauss_xy_middle), maxdiff_compute);
        ...
        maxdiff_compute =
          max13(abs(gauss_xy_pixels[ (x%3)*3 + cseymod3 ]
            - gauss_xy_middle), maxdiff_compute);
        ...
      }
    }
    ...
  }
}

```

distributivity :  $(x + 4) \% 3 = (x \% 3 + 4 \% 3) \% 3$   
 constant folding :  $= (x \% 3 + 1) \% 3$   
 constant unfolding :  $= (x \% 3 + 1 \% 3) \% 3$   
 invert distributivity :  $= (x + 1) \% 3$   
 (a)

modulo expansion:  $(x+2) \% 3 = 3 - x \% 3 - (x=1) \% 3$

(b)

FIG. 5

```

for (y=0; y<M+3; ++y) {
    cseymod3 = y%3;
    cseymin1mod3 = (y-1)%3;
    cseymin2mod3 = (y-2)%3;
    cseymin4mod3 = (y-4)%3;
    for (x=0; x<N+5; ++x) {
        ...
        if (x>=4 && x<=N+3 && y>=3 && y<=M+1) {
            if ((x-5)>=1 && (y-3)>=1) {
                if (out_compute == 255) {
                    cseymin4mod3x3 = ((x-4)%3)*3;
                    if (comp_edge_pixels [ csexmin4mod3x3 + cseymin2mod3 ] <comp_edge_middle) out_compute=0;
                    if (comp_edge_pixels [ csexmin4mod3x3 + cseymin4mod3 ] <comp_edge_middle) out_compute=0;
                    if (comp_edge_pixels [ ((x-5)%3)*3 + cseymin4mod3 ] <comp_edge_middle) out_compute=0;
                    ...
                }
            }
            if ((x-3)<=N-2 && (y-2)<= (M-2)) {
                csexmin2mod3x3 = ((x-2)%3)*3;
                maxdiff_compute =
                    max13 (abs (gauss_xy_pixels [ csexmin2mod3x3 + cseymin1mod3 ]
                        - gauss_xy_middle), maxdiff_compute);
                ...
                maxdiff_compute =
                    max13 (abs (gauss_xy_pixels [ csexmin2mod3x3 + cseymod3 ]
                        - gauss_xy_middle), maxdiff_compute);
                ...
                maxdiff_compute =
                    max13 (abs (gauss_xy_pixels [ (x%3)*3 + cseymod3 ]
                        - gauss_xy_middle), maxdiff_compute);
                ...
            }
        }
        ...
    }
}

```

FIG. 6

FIG. 7

```

cseymod3 = -1;
for (y=0; y<M+3; ++y) {
    cseymin1mod3 = cseymod3;
    cseymod3 = y%3;
    cseymin2mod3 = 3-cseymod3-cseymin1mod3;
    for (x=0; x<N+5; ++x) {
        ...
        if (x>=4 && x<=N+3 && y>=3 && y<=M+1) {
            if ((x-5)>=1 && (y-3)>=1) {
                if (out_compute == 255) {
                    csexmin4mod3x3 = ((x-4)%3)*3;
                    if (comp_edge_pixels [ csexmin4mod3x3 + cseymin2mod3 ] <comp_edge_middle) out_compute=0;
                    if (comp_edge_pixels [ csexmin4mod3x3 + cseymin1mod3 ] <comp_edge_middle) out_compute=0;
                    if (comp_edge_pixels [ ((x-5)%3)*3 + (cseymin1mod3) <comp_edge_middle) out_compute=0;
                    ...
                }
            }
            if ((x-3)<=N-2 && (y-2)<= (M-2)) {
                csexmin2mod3x3 = ((x-2)%3)*3;
                maxdiff_compute =
                    max13 (abs (gauss_xy_pixels [ csexmin2mod3x3 + cseymin1mod3 ]
                        - gauss_xy_middle), maxdiff_compute);
                ...
                maxdiff_compute =
                    max13 (abs (gauss_xy_pixels [ csexmin2mod3x3 + cseymod3 ]
                        - gauss_xy_middle), maxdiff_compute);
                ...
                maxdiff_compute =
                    max13 (abs (gauss_xy_pixels [ (x%3)*3 + cseymod3 ]
                        - gauss_xy_middle), maxdiff_compute);
                ...
            }
        }
        ...
    }
}

```

```

cseymod3 = -1;
for (y=0; y<M+3; ++y) {
    cseymmin1mod3 = cseymod3;
    cseymod3 = y%3;
    cseymmin2mod3 = 3-cseymod3-cseymmin1mod3;
    for (x=0; x<N+5; ++x) {
        ...
        if (x>=4 && x<=N+3 && y>=3 && y<=M+1) {
            csexmod3x3 = (x%3)*3;
            csexmin2mod3x3 = ((x-2)%3)*3;
            cseymmin4mod3x3 = ((x-4)%3)*3;
            csexmin5mod3x3 = ((x-5)%3)*3;
            if ((x-5)>=1 && (y-3)>=1) {
                if (out_compute == 255) {
                    if (comp_edge_pixels [ csexmin4mod3x3 + cseymmin2mod3 ] <comp_edge_middle) out_compute=0;
                    if (comp_edge_pixels [ csexmin4mod3x3 +cseymmin1mod3 ] <comp_edge_middle) out_compute=0;
                    if (comp_edge_pixels [ csexmin5mod3x3 +cseymmin1mod3 ] <comp_edge_middle) out_compute=0;
                }
            }
            if ((x-3)<=N-2 && (y-2)<= (M-2) ) {
                maxdiff_compute =
                    max13 (abs (gauss_xy_pixels [ csexmin2mod3x3 + cseymmin1mod3 ]
                        - gauss_xy_middle), maxdiff_compute);
                ...
                maxdiff_compute =
                    max13 (abs (gauss_xy_pixels [ csexmin2mod3x3 + cseymod3 ]
                        - gauss_xy_middle), maxdiff_compute);
                ...
                maxdiff_compute =
                    max13 (abs (gauss_xy_pixels [ csexmod3x3 + cseymod3 ]
                        - gauss_xy_middle), maxdiff_compute);
                ...
            }
        }
        ...
    }
}

```

FIG. 8



```

cseymod3 = -1;
for (y=0; y<M+3; ++y) {
    cseymin1mod3 = cseymod3;
    cseymod3 = y%3;
    cseymin2mod3 = 3-cseymod3-cseymin1mod3;
    for (x=0; x<N+5; ++x) {
        csexmod3x3 = (x%3)*3;
        csexmin2mod3x3 = ((x-2)%3)*3;
        cseymin4mod3x3 = ((x-4)%3)*3;
        csexmin5mod3x3 = ((x-5)%3)*3;
        ...
        if (x>=4 && x<=N+3 && y>=3 && y<=M+1) {
            if ((x-5)>=1 && (y-3)>=1) {
                if (out_compute == 255) {
                    if (comp_edge_pixels [ csexmin4mod3x3 + cseymin2mod3 ] <comp_edge_middle) out_compute=0;
                    if (comp_edge_pixels [ csexmin4mod3x3 +cseymin1mod3 ] <comp_edge_middle) out_compute=0;
                    if (comp_edge_pixels [ csexmin5mod3x3 +cseymin1mod3 ] <comp_edge_middle) out_compute=0;
                    ...
                }
            }
            if ((x-3)<=N-2 && (y-2)<= (M-2) ) {
                maxdiff_compute =
                    max13 (abs (gauss_xy_pixels [ csexmin2mod3x3 + cseymin1mod3 ]
                        - gauss_xy_middle), maxdiff_compute);
                ...
                maxdiff_compute =
                    max13 (abs (gauss_xy_pixels [ csexmin2mod3x3 + cseymod3 ]
                        - gauss_xy_middle), maxdiff_compute);
                ...
                maxdiff_compute =
                    max13 (abs (gauss_xy_pixels [ csexmod3x3 + cseymod3 ]
                        - gauss_xy_middle), maxdiff_compute);
                ...
            }
        }
        ...
    }
}

```

FIG. 10

```

cseymod3 = -1;
for (y=0; y<M+3; ++y) {
    cseymmin1mod3 = cseymod3;
    cseymod3 = y%3;
    cseymmin2mod3 = 3-cseymod3-cseymmin1mod3;
    csexmod3x3 = -3;
    for (x=0; x<N+5; ++x) {
        csexmin1mod3x3 = csexmod3x3;
        csexmod3x3 = (x%3)*3;
        csexmin2mod3x3 = 9-csexmod3x3-csexmin1mod3x3;
        ...
        if (x>=4 && x<=N+3 && y>=3 && y<=M+1) {
            if ((x-5)>=1 && (y-3)>=1) {
                if (out_compute == 255) {
                    if (comp_edge_pixels [ csexmin1mod3x3 + cseymmin2mod3 ] <comp_edge_middle) out_compute=0;
                    if (comp_edge_pixels [ csexmin1mod3x3 +cseymmin1mod3 ] <comp_edge_middle) out_compute=0;
                    if (comp_edge_pixels [ csexmin2mod3x3 +cseymmin1mod3 ] <comp_edge_middle) out_compute=0;
                    ...
                }
            }
            if ((x-3)<=N-2 && (y-2)<= (M-2) ) {
                maxdiff_compute =
                    max13 (abs (gauss_xy_pixels [ csexmin2mod3x3 + cseymmin1mod3 ]
                        - gauss_xy_middle), maxdiff_compute);
                ...
                maxdiff_compute =
                    max13 (abs (gauss_xy_pixels [ csexmin2mod3x3 + cseymod3 ]
                        - gauss_xy_middle), maxdiff_compute);
                ...
                maxdiff_compute =
                    max13 (abs (gauss_xy_pixels [ csexmod3x3 + cseymod3 ]
                        - gauss_xy_middle), maxdiff_compute);
                ...
            }
        }
        ...
    }
}

```

FIG. 11

```

cseymod3 = -1;
for (y=0; y<M+3; ++y) {
    cseymin1mod3 = cseymod3;
    cseymod3 = y%3;
    cseymin2mod3 = 3-cseymod3 - cseymin1mod3;
    cseymod3x3 = -3;
    cseymin1mod2 = (y-1)%2;
    cseymod2 = 1 - cseymin1mod2;
    for (x=0; x<N+5; ++x) {
        csexmin1mod3x3 = cseymod3x3;
        csexm3mod3x3 = (x%3)*3;
        csexmin2mod3x3 = 9-csexm3mod3x3 - csexmin1mod3x3;
        csexmin1x2 = (x-1)*2;
        csexmin3x2 = csexmin1x2-4;
        ...
        if (x>=3 && x<N+3 && y>=2 && y<=M+2)
            tmparray [ (csexmin3x2 + cseymod2) %160
                + (csexmin3x2 + cseymod2) / 160*256 + 96 ]
                = comp_edge_pixels [ csexm3mod3x3
                    + cseymin2mod3 ] = maxdiff_compute;
        ...
        if (x>=1 && x<N+1 && y>=1 && y<=M)
            tmparray [ (csexmin1x2 + cseymin1mod2) %64
                + (csexmin1x2 + cseymin1mod2) / 64*256 ]
                = gauss_xy_pixels [ csexmin1mod3x3
                    + cseymin1mod3 ] = gauss_xy_compute;
        ...
    }
}
}
}

```

FIG. 12

```

cseymod3 = -1;
for (y=0; y<M+3; ++y) {
  cseymin1mod3 = cseymod3;
  cseymod3 ++;
  if (cseymod3 >= 3) { cseymod3 -= 3; }
  cseymmin2mod3 = 3-cseymod3 - cseymin1mod3;
  cseymin1mod2 = (y-1) & 1;
  cseymod2 = 1 - cseymin1mod2;
  cseymod3x3 = -3;
  csexx2mod160_1_2 = cseymod2 - 8;
  csexx2div160_1_2 = 0;
  for (x=0; x<N+5; ++x) {
    csexmin1mod3x3 = csexmod3x3;
    csexmod3 ++;
    if (csexmod3 >= 3) { csexmod3 -= 3; }
    csexmod3x3 = csexmod3 * 3;
    csexmin2mod3x3 = 9 - csexmod3x3 - csexmin1mod3x3;
    csexx2mod160_1_2 += 2;
    csexmin1x2 = (x-1) * 2;
    csexmin3x2 = csexmin1x2 - 4;
    if (csexx2mod160_1_2 >= 160) { csexx2mod160_1_2 -= 160; csexx2div160_1_2 ++; }
    ...
  }
  if (x >= 3 && x < N+3 && y >= 2 && y < M+2)
    tmparray [ csexx2mod160_1_2 + csexx2div160_1_2 * 256 + 96 ]
      = comp_edge_pixels [ csexmod3x3
        + cseymin2mod3 ] = maxdiff_compute;

  if (x >= 1 && x < N+1 && y >= 1 && y <= M)
    tmparray [ ( (csexmin1x2 + cseymin1mod2) & 63)
      + ( (csexmin1x2 + cseymin1mod2) >> 6) * 256 ]
      = gauss_xy_pixels [ csexmin1mod3x3
        + cseymin1mod3 ] = gauss_xy_compute;
  ...
}
}
}
}

```

FIG. 13

